IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **DeWitt, Jr. et al.**	§	Group Art Unit: **2193**
Serial No.: **10/675,776**	§	Examiner: **Vu, Tuan A.**
Filed: **September 30, 2003**	§	Attorney Docket No.: **AUS920030481US1**
For: **Method and Apparatus for Counting Execution of Specific Instructions and Accesses to Specific Data Locations**	§	

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

PRELIMINARY AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

A fee of $810.00 is required for filing a Request for Continued Examination. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is necessary, I authorize the Commissioner to charge the necessary extension fees to Deposit Account No. 09-0447.

Prior to continued examination of this application, please amend the above-identified application as follows:

IN THE CLAIMS:

1. (Currently Amended) A method in a data processing system for monitoring execution of instructions, the method comprising:

receiving a bundle at an instruction cache, the bundle containing an instruction;

responsive to receiving the bundle, determining whether the bundle ~~an instruction~~ contains an indicator, wherein the indicator identifies the instruction ~~or a first memory location~~ as one that is to be monitored by a performance monitor unit; [[and]]

~~incrementing a counter associated with the instruction in response to detecting execution of the instruction and~~ responsive to a determination that the ~~instruction is associated with~~ bundle contains the indicator, incrementing a counter associated with the instruction wherein the incrementing ~~providing~~ provides a count of a number of times the instruction [[was]] is executed; and

sending the bundle from the instruction cache to a functional unit for execution of the instruction.

2. (Currently Amended) The method of claim 1 further comprising:

resetting the counter if the counter exceeds a threshold value; and reading a value of the counter prior to the counter exceeding the threshold value.

3-5. Canceled.

6. (Original) The method of claim 1, wherein the counter is located in a shadow memory.

7 – 25. Canceled.

26. (New) The method of claim 1 further comprising using a spare field in the bundle to contain the indicator.

27. (New) The method of claim 1, further comprising:

responsive to a determination that the bundle contains the indicator, sending a signal to the performance monitor unit.

28. (New) The method of claim 27, wherein the step of incrementing the counter associated with the instruction is performed by the performance monitor unit.

29. (New) The method of claim 1, further comprising:

responsive to a determination that the bundle contains the indicator, beginning incrementing the counter, wherein the counter tracks any subsequent instruction executed by an associated processor.

30. (New) The method of claim 29 wherein the bundle is a first bundle, the method further comprising:

receiving a second bundle at the instruction cache;

responsive to receiving the second bundle, determining whether the second bundle contains a second indicator; and

responsive to a determination that the second bundle contains the second indicator, ending incrementing the counter.

31. (New) The method of claim 30, wherein the counter and the second counter are identical.

32. (New) A computer program product comprising:

a computer readable medium having computer useable program code for monitoring execution of instructions, the computer program product comprising:

computer usable program code for receiving a bundle at an instruction cache, the bundle containing an instruction;

computer usable program code for, responsive to receiving the bundle, determining whether the bundle contains an indicator, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;

computer usable program code for, responsive to a determination that the bundle contains

the indicator, incrementing a counter associated with the instruction wherein the incrementing provides a count of a number of times the instruction is executed; and

computer usable program code for, sending the bundle from the instruction cache to a functional unit for execution of the instruction.

33. (New) The computer program product of claim 32 further comprising:

computer usable program code for resetting the counter if the counter exceeds a threshold value; and

computer usable program code for reading a value of the counter prior to the counter exceeding the threshold value.

34. (New) The computer program product of claim 32, wherein the computer usable program code for incrementing the counter further comprises computer usable program code for incrementing the counter, wherein the counter is located in a shadow memory.

35. (New) The computer program product of claim 32 further comprising using a spare field in the bundle to contain the indicator.

36. (New) The computer program product of claim 32, further comprising:

computer usable program code, responsive to a determination that the bundle contains the indicator, for sending a signal to the performance monitor unit.

37. (New) The computer program product of claim 36, wherein the computer usable program code for incrementing the counter associated with the instruction computer usable program code to be executed on the performed by the performance monitor unit.

38. (New) The computer program product of claim 32, further comprising:

computer usable program code, responsive to a determination that the bundle contains the indicator, for beginning incrementing the counter, wherein the counter tracks any subsequent instruction executed by an associated processor.

39. (New) The computer program product of claim 38 wherein the bundle is a first bundle, the computer program product further comprising:

computer usable program code for receiving a second bundle at the instruction cache;

computer usable program code, responsive to receiving the second bundle, for determining whether the second bundle contains a second indicator; and

computer usable program code, responsive to a determination that the second bundle contains the second indicator, for ending incrementing the counter.

40. (New) The computer program product of claim 39, wherein the counter and the second counter are identical.

41. (New) A data processing system comprising:

a bus;

a communications unit connected to the bus;

a storage device connected to the bus, wherein the storage device includes computer usable program code; and

a processor unit connected to the bus, wherein the processor unit executes the computer usable program code to receive a bundle at an instruction cache, the bundle containing an instruction, responsive to receiving the bundle, to determine whether the bundle contains an indicator, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit, responsive to a determination that the bundle contains the indicator, to increment a counter associated with the instruction wherein the incrementing provides a count of a number of times the instruction is executed, and

to send the bundle from the instruction cache to a functional unit for execution of the instruction.

42. (New) The data processing system of claim 41, wherein the processor unit further executes computer usable program code to reset the counter if the counter exceeds a threshold value, and to read a value of the counter prior to the counter exceeding the threshold value.

43. (New) The data processing system of claim 41, wherein the processor unit executing computer usable program code to increment the counter further comprises executing computer usable program code to increment the counter, wherein the counter is located in a shadow memory.

44. (New) The data processing system of claim 41 wherein the processor unit further executes computer usable program code to use a spare field in the bundle to contain the indicator.

45. (New) The data processing system of claim 41, wherein the processor unit further executes computer usable program code, responsive to a determination that the bundle contains the indicator, to send a signal to the performance monitor unit.

46. (New) The data processing system of claim 45, the computer usable program code for incrementing the counter associated with the instruction computer usable program code to be executed on the performed by the performance monitor unit.

47. (New) The data processing system of claim 41, wherein the processor unit further executes computer usable program code, responsive to a determination that the bundle contains the indicator, to begin incrementing the counter, wherein the counter tracks any subsequent instruction executed by an associated processor.

48. (New) The data processing system of claim 47 wherein the bundle is a first bundle, wherein the processor unit further executes computer usable program code to receive a second bundle at the instruction cache, responsive to receiving the second bundle, to determine whether the second bundle contains a second indicator, and responsive to a determination that the second bundle contains the second indicator, to end incrementing the counter.

REMARKS

These amendments were made to clarify the presently claimed subject matter from the cited reference. No new matter has been added by these amendments. Claims 1and 2 have been amended. Claims 3-5, and 7-25 have been canceled. Claims 26-48 have been added. Support for the new claims can be found on pages 26-32 of the original as-filed application.

I. Double Patenting

I.A. Claims 1, 13, and 21

The Examiner has provisionally rejected claims 1, 13, and 21 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 3, 20, and 25 of co-pending Application No. 10/675,777 (hereinafter '777).

Applicants acknowledge that this is a provisional rejection. As of the date of this response, neither the rejected claims, nor the claims of the co-pending application have been allowed. Therefore a terminal disclaimer would be premature at this point. Should the Examiner indicate that either the rejected claims, or the reference be allowable, a terminal disclaimer will be filed at that time.

I.B. Claims 1, 13, and 21

The Examiner has provisionally rejected claims 1, 13, and 21 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 9, 19, 23, and 25 of co-pending Application No. 10/675,778 (hereinafter '778).

Applicants acknowledge that this is a provisional rejection. As of the date of this response, neither the rejected claims, nor the claims of the co-pending application have been allowed. Therefore a terminal disclaimer would be premature at this point. Should the Examiner indicate that either the rejected claims, or the reference be allowable, a terminal disclaimer will be filed at that time.

I.C. Claims 1, 13, and 21

The Examiner has provisionally rejected claims 1, 13, and 21 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 9, and 17 of

co-pending Application No. 10/675,721 (hereinafter '721).

Applicants acknowledge that this is a provisional rejection. As of the date of this response, neither the rejected claims, nor the claims of the co-pending application have been allowed. Therefore a terminal disclaimer would be premature at this point. Should the Examiner indicate that either the rejected claims, or the reference be allowable, a terminal disclaimer will be filed at that time.

I.D. Claims 1, 10, 13, 18, 21 and 25

The Examiner has provisionally rejected claims 1, 10, 13, 18, 21 and 25 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 2 and 16 of co-pending Application No. 10/675,872 (hereinafter '872).

Applicants acknowledge that this is a provisional rejection. As of the date of this response, neither the rejected claims, nor the claims of the co-pending application have been allowed. Therefore a terminal disclaimer would be premature at this point. Should the Examiner indicate that either the rejected claims, or the reference be allowable, a terminal disclaimer will be filed at that time.

I.E. Claims 10, 18, and 25

The Examiner has provisionally rejected claims 10, 18, and 25 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 9, and 17 of co-pending Application No. 10/682,385 (hereinafter '385).

Applicants acknowledge that this is a provisional rejection. As of the date of this response, neither the rejected claims, nor the claims of the co-pending application have been allowed. Therefore a terminal disclaimer would be premature at this point. Should the Examiner indicate that either the rejected claims, or the reference be allowable, a terminal disclaimer will be filed at that time.

II. Claim Objections

The Examiner has objected to claims 1, 13, and 21 as containing the phrase "contains an indicator." The Applicants have amended claim 1, overcoming the Examiner's rejection. Claims 13 and 21 have been canceled.

III. Claim Rejections 35 U.S.C. §112

The Examiner has rejected claims 1-9, 13-17, and 21-24 under 35 U.S. §112 as being indefinite. Claim 1 has been amended, overcoming the Examiner's rejection. Claims 2 and 6 depend from claim 1. Thus the rejection to those claims should also be overcome by the amendments to claim 1. The remainder of the objected claims have been canceled.

IV. Claim Rejections 35 U.S.C. §102(b)

The examiner has rejected claims 1-25 under 35 U.S.C. § 102 as being anticipated by U.S. Patent No. 7,752,062 to *Gover* et al. (hereinafter "*Gover*"). This rejection is respectfully traversed.

With regard to claim 1, the Examiner states:

> As per claim 1, *Gover* discloses a method in a data processing system for monitoring execution of instructions, the method comprising: determining whether an instruction contains an indicator (e.g., Fig. 5; count number, bit fields, MMCRO – col. 10, lines 31-35; Fig. 6A; col. 11, line 62 to col. 12, line 42) wherein the indicator identifies the instruction or first memory location as one that is to be monitored by a performance monitor unit (e.g., PMCn 51-Fig. 4, 6-6); and
>
> incrementing a counter associated with the instruction (e.g. even… to be recorded/counted, counter… selection, counter freeze – col. 10, lines 53-63; col. 11, lines 14-50) in response to detecting execution of the instruction and to a determination that the instruction is associated with the indicatory, the incrementing providing a count of a n8umer of times the instruction was executed (e.g. Fig. 3: FINISHED; number… branches dispatched… completed – col. 20, lines 48-65; load or store – col. 22 lines 10-35; Fig. 8).

Final Office Action dated August 1, 2007, pp. 7-8.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case each and every feature of the presently claimed invention is not identically

shown in the cited reference, arranged as they are in the claims.

Gover describes an example of a trace technique similar to those known to the applicant and described as prior art on page 4 of the application as filed. These trace techniques are described as:

> periodically sampling a program's execution flows to identify certain locations in the program in which the program appears to spend large amounts of time. This technique is based on the idea of periodically interrupting the application or data processing system execution at regular intervals, so-called sample-based profiling. At each interruption, information is recorded for a predetermined length of time or for a predetermined number of events of interest. For example, the program counter of the currently executing thread, which is an executable portion of the larger program being profiled, may be recorded during the intervals. These values may be resolved against a load map and symbol table information for the data processing system at post-processing time, and a profile of where the time is being spent may be obtained from this analysis.

Specification, p. 4, ll. 3-19.

As described in *Gover*:

> a history of events is gathered in a processing system. The historical data is collected in a manner that is noninvasive to the system's operation but occurs within the processor. Thus, the data is unbiased and unaffected by external test instruments, while obtaining a cycle by cycle history of events during processing. Further, a straightforward manner of specifically choosing the instructions that initiate and complete monitoring activity is also obtained, which is normally more difficult in a processing system.

Gover, col. 3, ll. 44-52.

> Performance monitor 50, in a preferred embodiment, is a software-accessible mechanism intended to provide detailed information with significant granularity concerning the utilization of PowerPC instruction execution and storage control. Generally, the performance monitor 50 includes an implementation-dependent number (e.g., 2-8) of counters 51, e.g, PMC1-PMC8, used to count processor/storage related events.

Gover, col. 8, ll. 19-26.

> In operation, a notification signal is sent to PM 50 from time base facility 52 when a predetermined bit is flipped. The PM 50 then saves the machine state values in special purpose registers. In a different scenario, the PM 50 uses a "performance monitor" interrupt signalled by a negative counter (bit zero on or "1") condition.

Gover, col. 9, ll. 21-26.

In support of the current rejection, the Examiner cites the following section from *Gover*:

> Further for those events being monitored that are time sensitive, e.g., a number of stalls, idles, etc., the count number data is collected over a known number of elapsed cycles, so that the data has a context in terms of a sampling period.

Gover, col. 10, ll. 31-35.

This section states that certain monitored events are collected in a known period of time. This allows time sensitive events to be normalized over the collected time period to ease comparison of the events. This cited section does not state that an instruction is associated with an indicator, as recited in claim 1. Instead, *Gover* teaches that the monitor mode control register is incremented based on the occurrence of an event. A register is not an instruction.

In an attempt to expedite prosecution, claim 1 has been amended to incorporate additional features found in the specification. Applicants do not concede that the originally filed claims are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are included only to facilitate expeditious prosecution. Applicants respectfully reserve the right to pursue these and other claims in one or more continuations and/or divisional patent applications.

Claim 1, as amended is as follows:

> 1. A method in a data processing system for monitoring execution of instructions, the method comprising:
>
> receiving a bundle at an instruction cache, the bundle containing an instruction;
>
> responsive to receiving the bundle, determining whether the bundle contains an indicator, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;
>
> responsive to a determination that the bundle contains the indicator, incrementing a counter associated with the instruction wherein the incrementing provides a count of a number of times the instruction is executed; and
>
> sending the bundle from the instruction cache to a functional unit for execution of the instruction.

Claim 1 now recites additional limitations that are not found within the *Gover* reference. Applicants submit that in light of the amendments to the claims, that the Examiner's previous rejections are now moot. *Gover* does not anticipate the claims as amended for at least the reason that *Gover* does not disclose the additional features of amended claim 1.

New claims 32 and 41 contain features similar to those found in amended claim 1. Therefore, the Applicants respectfully submit that *Gover* does not anticipate the newly added claims either, for at least the reason that *Gover* does not disclose the additional features of amended claim 32 and 41 not presented in previous claims.

V. Conclusion

It is respectfully urged that the subject application is patentable over *Gover* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: November 1, 2007

Respectfully submitted,

/Brandon G. Williams/

Brandon G. Williams
Reg. No. 48,844
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants